CS 201 Homework 02
CHANGEME: CS201 Homework 02

Stefano Fochesatto

September 24, 2019

Source Code Link: `https://github.com/StefanoFochesatto/cs201/tree/master/Homeworks/HW2`

This homework took approximately 05 hours to complete.

1 Design

For the main names program we want to make sure to use string vectors and functions to simplify and componentize our source code. This approach proved extremely helpful when writing later programs, because it made it simple and clear to copy and paste similar functions between programs. The Names program has four different functions, PrintNames, DoesNameExist, InputNames, and SortAlphibetical. For the most part they are all pretty simple, and it all starts with initializing the names vector. Then the user inputs the vector's strings using the InputNames function, which essentially consists of a for loop that iterates on the index of the vector. In retrospect, since the length was fixed by the for loop it could have been more effective to use a string array, or use a while loop with a termination input. them the names vector it taken through the DoesNameExist function which uses a fixed while loop to iterate through the index of the vector and an if statement to check if the name at the index is equal to the searched for name. Then the vector is taken through the PrintNames function which just prints out each element in the vector with a for loop. Then The vector is organised by alphabetical order using the SortAlphibetical function. Would be cool to implement some sort of menu system like in the scores program to actually let the user decide what functions to call.

2 Post Mortem

What I found to be very difficult throughout this homework was deciding which parts of the source code to put into functions. There would be cases where i would want to put a whole section of my source code into a function just to make everything neater, but it hardly made any sense at all because the use case would have to be so specific I'd hardly ever have to use it again i.e the menu in the scores program. Furthermore I learned a lot about std::pairs documentation when working on that scores program. I decided to make a vector¡std::pairs¿ instead of two separate vectors for the score and names. This lead to very streamlined code but it took too long to learn and I didn't have time to implement some of the program requirements, i.e error for repeated names.

3 Answers to Questions

In this section, you will write the answers to the questions in the homework assignment.

- What are the typical sizes for a char, int and a double? Char and int are typically one byte, although signed ints can go up to 64 bits with 128 bit support coming. A double is typically 8 bytes.

- What is a definition?

- What is an initialization and how does it differ form an assignment?

Initialization tells the compiler to allocate memory for whatever variable, while assignment simply assigns a value to the variable. For example initialization looks like this, int i = 7; or like this int i; while assignment looks like, i = 7;

- Define a rule to help decide if conversion from one type to another is safe.
  Only convert form one type to another if the conversion is a promotion, i.e a value in a smaller variable is assigned to a larger variable with no loss of data. Conversions like taking an int or float to a double. Even a bool or char to any other built-in type.

- What is a computation?
  A computation is the process of taking an input and converting it into a useful output.

- What is the difference between a statement and an expression?
  An expression computes a value from a number of operands, while statements are used more generally to do the rest of the thing we need to do when programming. Consider, int n = 1; this is a declaration statement, while n = n - 1; is an expression statement.

- What is a a constant expression?
  Symbolic constant, that is, a named object to which you can't give a new value after it has been initialized. Consider pi, e or C. We can make our own constants by declaring them, consider; const int n = 1;.

- What can you do with an int that you can't do with a string?

- What can you do with a string that you cannot do with an int?

- What does vector¡char¿ alphabet(26) do?
  It initializes a character vector named alphabet size 26¿

4 Sample Output

Listing 1: Sample Program Output

```
Please enter a name: Stefano
Please enter a name: Brons
Please enter a name: Rachel
Please enter a name: Liz
Please enter a name: Deseree
Please enter a name: Rohan
Please enter a name: Dalen
Please enter a name: Jill
Please enter a name: Sarah
Please enter a name: Erik
Type a name to search: Liz
1
Stefano Brons Rachel Liz Deseree Rohan Dalen Jill
   Sarah Erik
Brons Dalen Deseree Erik Jill Liz Rachel Rohan Sarah
   Stefano
Program ended with exit code: 0
```

5 names

```
//
//  main.cpp
//  names
//
//  Created by Stefano Fochesatto on 9/20/19.
//  Copyright © 2019 Stefano Fochesatto. All rights reserved.
// This program creates a list of names size 10, searches, prints, and sorts

#include <iostream>
```

```cpp
#include <vector>
#include <string>
#include <algorithm>
#include "InputNames.hpp"
#include "DoesNameExist.hpp"
#include "PrintNames.hpp"
#include "SortAlphabetical.hpp"

std::vector<std::string> InputNames(std::vector<std::string> names);
bool DoesNameExist(const std::string nameToFind, std::vector<std::string> names);
void PrintNames(std::vector<std::string> names);
std::vector<std::string> SortAlphibetical(std::vector<std::string> names);


int main(int argc, const char * argv[]){
    std::vector<std::string> names;
    names = InputNames(names);


    std::string nameToFind;
    std::cout <<"Type a name to search: ";
    std::getline(std::cin, nameToFind);
    std::cout << DoesNameExist(nameToFind,names)<< "\n";



    PrintNames(names);

    names = SortAlphibetical(names);

    PrintNames(names);

}
```

6 Sample Output: Money

Listing 2: Sample Program Output

```
How many pennies: 1
How many nickles: 5
How many dimes: 2
How many quarters: 3
How many half-dollars: 1
You have 1 penny.
You have 5 nickles.
You have 2 dimes.
You have 3 quarters.
You have 1 half-dollar.
Your total is: $1.71
Program ended with exit code: 0
```

7 Money

```
//
//  Coin_Math.hpp
//  Money
//  This program counts change.
//
//  Created by Stefano Fochesatto on 9/23/19.
//  Copyright © 2019 Stefano Fochesatto. All rights reserved.
//

#ifndef Coin_Math_hpp
#define Coin_Math_hpp

#include <iostream>
#include <vector>
#include <functional>
#include <numeric>
#include <stdio.h>
int Coin_Math(std::vector<int> Coins){

    int coinVal[] = {1,5,10,25,50};

    std::vector<int> CoinsXCoinVal;

    for (int i = 0 ; i<=4; i++){
    int TotalCoinVal;
        TotalCoinVal = Coins.at(i)*coinVal[i];
        CoinsXCoinVal.push_back(TotalCoinVal);
       }
    return std::accumulate(CoinsXCoinVal.begin(), CoinsXCoinVal.end(), 0);
}


#endif /* Coin_Math_hpp */
```

8 Sample Output: scores

Listing 3: Sample Program Output

```
Please enter a Name and then a Score:Stefano
100
Please enter a Name and then a Score:Brons
```

```
100
Please enter a Name and then a Score:Luca
50
Please enter a Name and then a Score:NoName
0
Please select from the menu below,
1: Add More Names and Scores
2: Print the Names and Scores
3: Search for a Name
4: Search for a Score
2
Brons, 100
Luca, 50
Stefano, 100
Please select from the menu below,
1: Add More Names and Scores
2: Print the Names and Scores
3: Search for a Name
4: Search for a Score
4
Enter the score you want to search: 100
Brons, 100
Stefano, 100
Please select from the menu below,
1: Add More Names and Scores
2: Print the Names and Scores
3: Search for a Name
4: Search for a Score
7
Exiting program!,
Program ended with exit code: 0
```

9 scores

```cpp
//
//  main.cpp
//  Scores
//
//  Created by Stefano Fochesatto on 9/23/19.
//  Copyright © 2019 Stefano Fochesatto. All rights reserved.
//

#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
#include "Input_Scores.hpp"
#include <utility>
int main(int argc, const char * argv[]) {
    std::vector<std::pair<std::string, int>> Names_Scores;
    Names_Scores = Input_Scores(Names_Scores);






    bool MenuRuntime = true;
    while (MenuRuntime == true){
    int MenuSelection;
    std::cout << "Please select from the menu below,\n";

    std::cout << "1: Add More Names and Scores \n";
    std::cout << "2: Print the Names and Scores \n";
    std::cout << "3: Search for a Name \n";
    std::cout << "4: Search for a Score\n";
    std::cin >> MenuSelection;
    if (MenuSelection == 1){
    Names_Scores = Input_Scores(Names_Scores);
    }
    else if (MenuSelection == 2){
        for(int i = 0; i < Names_Scores.size(); i++)
        {
             std::cout << Names_Scores[i].first << ", " << Names_Scores[i].second <<
        }
    }
    else if (MenuSelection == 3){
        std::string nameToFind;
        std::cout << "Enter the name you want to search: ";
        std::cin >> nameToFind;
        int i = 0;
        int counter=0;
        while (i <= Names_Scores.size()){
                std::string cName = Names_Scores[i].first;
                if (cName == nameToFind){
                     std::cout << Names_Scores[i].first << ", " << Names_Scores[i].s
                    counter++;
                }
                i++;
            }
        if (counter==0){
            std::cout <<"Unable to find Name!! \n";

        }
    }
    else if (MenuSelection == 4){
        int scoreToFind;
        std::cout << "Enter the score you want to search: ";
        std::cin >> scoreToFind;
        int i = 0;
        int counter=0;
        while (i <= Names_Scores.size()){
                int cScore = Names_Scores[i].second;
                if (cScore == scoreToFind){
                     std::cout << Names_Scores[i].first << ", " << Names_Scores[i].s
                    counter++;
                }
                i++;
            }
        if (counter==0){
            std::cout <<"Unable to find Score!! \n";

        }

    }
    else{
        std::cout << "Exiting program!,\n";
        MenuRuntime = false;

    }

    }
}
```